Webology (ISSN: 1735-188X)
Volume 18, Number 4, 2021

Engineering Based On Stacking And Features Handling Data Imbalance With Semi-Supervised Intrusion Detection

A.Sagaya Priya*, S.Britto Ramesh Kumar**

* Research scholar, Department of Computer Science, St.Joseph's College(Autonomous), .Affiliated to Bharathidasan University, Trichy, Tamilnadu, India.

**Assistant Professor, Department of Computer Science, St.Joseph's College(Autonomous), .Affiliated to Bharathidasan University, Trichy, Tamilnadu, India.

Abstract

When there is a disparity between the amount of data in each category, Intrusion Detection Systems are profoundly affected. While models can handle some degree of imbalance directly, greater disparities have significant effects on simulation results. The degree of imbalance in a network transaction varies greatly depending on the network in question. In order to deal with varying degrees of data imbalance in a network, this paper proposes a unified framework for doing so. The proposed Stacking and Feature engineering based Semi supervised (SFS) model presents a combined architecture that integrates data balancing, feature engineering, and a stacking based prediction model, all of which work together to correct data imbalance, shrink data size, and make accurate predictions. The data is evenly distributed using an oversampling technique, and the problem of overtraining caused by oversampling is addressed via a stacking architecture. In order to illustrate the generalizability of the SFS model, several imbalanced intrusion detection datasets have been explored. The results of the experiments and comparisons show that the system performs well in general, and especially well at predicting the minority classes.

Keywords: Intrusion Detection; Data Imbalance; Stacking; Feature Engineering; Oversampling; Semi Supervised Learning.

1.Introduction

Today's civilization can't function without the Internet and the various forms of communication made possible by the web. As a result, cyber security dangers have increased significantly. There has been a rise in recent years in the frequency, severity, and complexity of attacks on our cyber defences. Most businesses have shifted to online processing [1], which has coincided with the rapid spread of COVID 19. However, while internet processing has given people more freedom, it has also given cybercriminals access to a wealth of sensitive data. Cyberattacks on businesses have been rising steadily and hit a peak in April 2021 [2].

By putting the spotlight on potential security vulnerabilities, this scenario has helped to detect breaches and maintain a secure network [3]. Intrusion detection systems are a method for

inspecting network packets for unusual activity. There are three main kinds of intrusion detection systems [4]: host-based systems, network-based systems, and hybrid systems that use both. In order to identify intrusions, host based solutions just require a single machine to function. Intrusion detection systems that operate on the network level monitor data packets for irregularities [5]. System components for host and network intrusion detection are combined in hybrid intrusion detection systems.

In today's digitally connected world, automatic intrusion detection has become a need [6, 7]. Needed features of an intrusion detection system include rapid detection rates, high accuracy, and low computational complexity. Due to the fast pace at which network packets move, intrusion detection must be quick and require minimal processing resources [8]. It is challenging to find a single model that can meet all three needs; instead, each model has its own set of trade-offs that must be considered in light of the context in which it will be used.

Network intrusion detection typically employs supervised learning techniques, such as classification models [9]. Classification performance is severely impacted by problems like data imbalance and noise in the network data. Complexity arises when attempting classification on unbalanced data because of the inherent bias introduced by the unbalance [10]. The level of bias generated during classification is also heavily influenced by the degree of imbalance. Data collected in real time from a network is typically skewed. So, it's important to have a model that works well with real-time data and can accommodate different types of imbalance [11].

This paper introduces the Semi supervised (SFS) machine learning model for intrusion detection in a networked setting, which is based on stacking and feature engineering. Additionally, the method employs a balancing module to deal with the disproportion. By compressing the data using feature engineering, we can make the model's computations more efficient. Improved forecasts can be obtained by combining the stacking method with semi-supervised learning.

2. Related Works

The increasing importance of the information being transferred via networks has led to an increase in the frequency and severity of assaults on those networks. The state-of-the-art in network intrusion detection is presented here.

Yerriswamy et al. present a unified strategy that combines genetic-based grey wolf optimization methods with a feature selection algorithm. [12]. An improved prediction model for network intrusion detection is developed in this study by enhancing the existing grey wolf optimization technique and integrating it with a genetic algorithm. Baklini et al. offer a DDoS-specific intrusion prevention system. [13]. In order to effectively detect DDoS attacks, this work combines a sliding window approach with a logical fractal dimension. The model computes the window size automatically and tests the accuracy of its predictions with varying values of hyper parameters to demonstrate its detection prowess. Behal et al. offer a time-window based method for mitigating DDoS attacks. [14]. This method distinguishes between normal and suspicious traffic by calculating the Shannon entropy of each packet. Models by

Jun et al. [15] and David et al. [16, 17] also make use of entropy-based approaches for intrusion detection. [16].

It has been proposed by Guarascio et al. [17] that a collaborative model be used to identify network intrusions. The primary focus of this effort is on developing cooperative detection models that will lead to an enhanced intrusion detection system as a whole. It specifies a system that facilitates the exchange of information in order to boost the reliability of predictions. Structured Threat Information CybereXpression (STIX) by Jordan et al. [18], Cyber Observable eXpression (CybOX) by Darley et al. [19], and Trusted Automated eXchange of Indicator Information (TAXII) by Darley et al. [20] are all related terms that deal with providing specifications for collaborative intrusion detection. The work of Prasad et al. [21] proposes a safe intrusion detection solution for MANET.

Pampapathi et al. [22] suggested a deep learning-based model for intrusion detection. In order to detect intrusions, this work combines a filtered deep learning model with a data communication strategy. These computations are carried out by the cluster heads in a model that is based on clustering. Additional publications by Siddiqui et al. [23] and Khanan et al. [24] that use clustering-based methods for intrusion detection are also worth looking into. In [25], the authors suggest a deep learning model for network intrusion detection that makes use of regularisation best auto encoder. Asif et al. [26] developed a MapReduce-based model for intrusion detection, and [27] used a feature engineering-based model. Our studies centre on the security concerns in the RPL and the potential attacks that could damage IoT equipment. The RPL protocol has been the target of numerous potential routing attacks.

3. **Stacking-and-feature-engineering-based Semi-supervised IDS**

Multiclass data and data imbalance in network transactions hamper intrusion detection. This paper proposes a classification architecture for imbalanced data. Combining stacking with feature engineering, the proposed architecture leverages semi-supervised prediction for training. The suggested model, Stacking and Feature Engineering based Semi supervised (SFS), has four modules. The first module does data pre-processing and balancing, followed by feature selection, semi-supervised first level predictions, and the final prediction. Here's the SFS algorithm.

Input: Imbalanced data (KDD CUP 99, NSL-KDD, UNSW-NB15)

Output: Predictions on imbalanced data

1) Enter transmission information from the network

Encoding and cleaning the data for consistency before using it

Determine the severity of the imbalance.

Choose two more minority records at random for each additional majority record in the data set.

b. Produce a fresh instance by averaging the values of the sampled ones.

Determine the feature entropy values using a decision tree 5.

To choose features based on entropy (step 6)

7 Use sampling with replacement to divide your data into several groups.

8 Make several versions of a model using a Gaussian mixture or a decision tree.

9. Provide a unique training dataset to each model you produce. a.

10: Give the trained models all of the training data to make predictions.

11, Combine predictions and labels to produce level 2 training data

12 - Train a Logistic Regression model with level 2 training data

13. For each instance I in the test data, we first a. send I to all the base learners, b. integrate the predictions, c. send the integrated predictions to the trained Logistic Regression model, and d. obtain final predictions.

3.1 The Pre-processing and Balancing of the Data

The proposed SFS model's efficacy and generalizability are evaluated using intrusion detection data from several datasets as training data. Features extracted from network traffic make up intrusion detection data. Improving the quality of the training data necessitates analysis of these aspects. Based on our examination, we know that the data has both categorical and numerical qualities in addition to string ones. While machine learning models can make direct use of numerical features, they should first analyse category and string qualities before employing them. It is common practise to use encoding methods to transform categorical attributes into numeric ones. In this work, one hot encoding is favoured. We get rid of the string properties. The class attribute is represented as a categorical field in some data sets. This property indicates whether the transmission was typical or not. Depending on the data collection, this value may be represented as a number of classes, each of which describes a different kind of unexpected traffic. Therefore, the multi class data is transformed into binary class data since the classification method is assumed to be binary classification in this work. The label encoding process is used on the class attribute to make it numerical.

The data used for intrusion detection is often skewed. Typical traffic patterns are represented by a huge number of records. However, evidence of incursion traffic in the form of logs is scarce. Rare events like these are extremely unusual. Due to its inherent low quality, this data typically yields poorer prediction results. As a result, this study employs an oversampling strategy to ensure that the data is representative and of high quality. The first step is to count how many new records need to be added to achieve statistical parity. Each newly minted record is a product of a union of two preexisting ones. Despite the fair distribution of records in the training set, oversampling typically leads to data overtraining because of the numerous nearly-

identical records. In order to solve this problem, the proposed model employs a sampling strategy.

3.2 Selection of Feature

Network information includes details about the packet's destination and the sort of network it travelled over. It generates a high number of features, which in turn leads to the curse of dimensionality. Oversampling occurs during the balancing phase and also increases the total number of instances. The sum of the training data grows as a result of these operations. Therefore, a feature selection strategy is used into this study to speed up the computation. One such model is a tree-based one for selecting features. The developed model is a meta transformer that measures entropy to determine which characteristics are most important. In order to determine which features are most important, a decision tree algorithm is used to the training data. The size of the training data is lowered using this procedure, which in turn requires less processing power.

3.3 **Prediction at the First Level with Partial Supervision**

Many different types of heterogeneous models are used at the first level of semi supervised prediction. The foundational prediction architecture was constructed using a mix of supervised and unstructured models. To provide machine learning models with a sufficient amount of training data, it is first split into many overlapping groups. In this study, we employ a hybrid of the Gaussian Mixture model and the Decision Tree model to accomplish our machine learning goals.

The Gaussian mixture model is a type of unsupervised clustering that works under the assumption that the data being clustered is normally distributed. The model creates a cluster out of data points that all share the same distribution. These models are probabilistic in nature, and they group data points in a manner similar to the "soft clustering" method. The main benefit of adopting the machine mixing model is that the clusters are determined by taking into account the current variance level in points. So, the likelihoods of a given point's membership in a certain cluster can be calculated using Gaussian mixture models.

The decision tree is a modelling tool that uses tree structures to generate branches according to the entropy values in the training data. In a decision tree, each node represents a condition and each branch indicates a possible course of action. The final forecast is displayed as leaf nodes. Especially though decision trees are a relatively simple learning model, they are still capable of dealing with changing data and making accurate predictions, even in a streaming environment.

These models are replicated many times, with each instance receiving a unique portion of the training data. Each model is tailored to use a unique subset of the full training dataset. By doing so, we mitigate overfitting brought on by excessive sampling. Once the training phase is complete, the training data is used to make initial predictions. All of these forecasts are combined to create the second-level stacking model's training data. This information is tagged with the class label and sent on to the next processing stage.

3.4 Final Prediction Made Using Stacking on the Second Level

The meta-model at the second level of the stacked-model hierarchy is trained with the help of the predictions made at the first level. Since this model draws from historical predictions rather than training data, it is thought to be more resilient to issues like noise that are typically present in real-time data. The most common meta-model employed here is logistic regression.

The statistical analysis method of logistic regression net predicts a binary result based on the provided training data. Analyzing the interdependencies between already independent variables allows for the making of predictions. By fitting the model onto a curve, it estimates the logistic model's parameters. Logistic regression is employed as the second-stage model since it relies on the predictions rather than the training data. The Logistic regression model is trained with the prediction data that was available in the previous step.

Level one models receive the test data and semi-supervised models' predictions, which are then combined with the test data and sent on to the logistic regression model. The predictions made by the logistic regression model will be used.

4. Results and Findings

Python has been used to implement the proposed Stacking and Feature engineering based Semi supervised (SFS) model. The KDD CUP 99 dataset, the NSL- KDD dataset, and the UNSW-NB15 dataset have all been used in SFS model analyses. Levels of noise and unevenness in each data set are different.

Figure 1 displays the SFS model's precision and recall (PR) curve over all three datasets. Highly efficient classifier models have both a high precision and recall. The graph displays recall percentages higher than 90% and precision percentages close to 1. This demonstrates the model's generic character and its strong capacity for making accurate predictions across a wide range of datasets with high production efficiency.

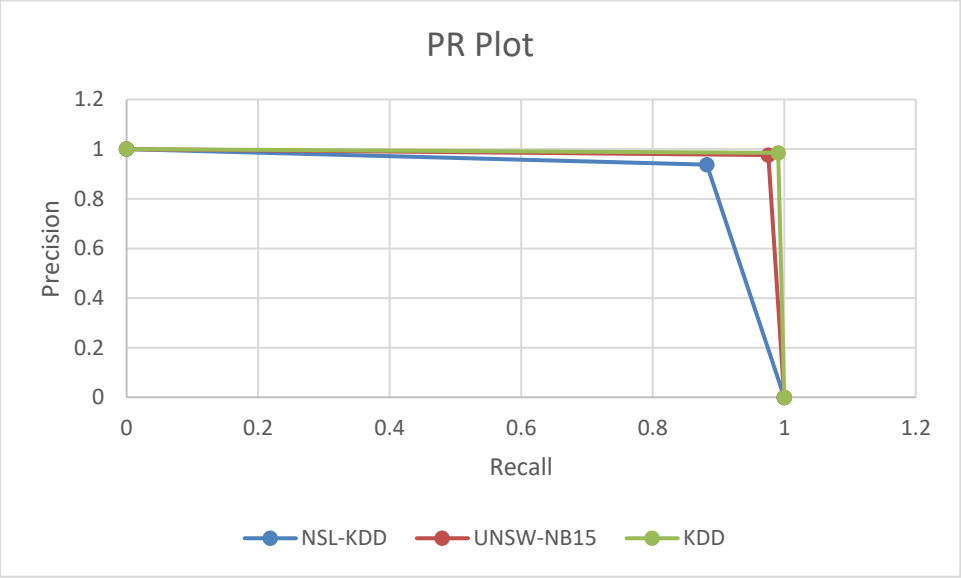

Figure 1:PR plot of SFS

Figure 2 displays a comparison of the AUC, F-measure, and aggregate measure accuracy. All three metrics offer an overall performance level that may be used to determine the model's performance in predicting over the binary class data, and they are calculated by combining the current performance metrics. Above-average levels of accuracy, F-measure, and area under the curve (AUC) may be shown in the chart for each of the three datasets. This result shows that the model is not biassed and can make accurate predictions in the aggregate. The model's objectivity demonstrates that it is unaffected by the degree of imbalance present in the data.

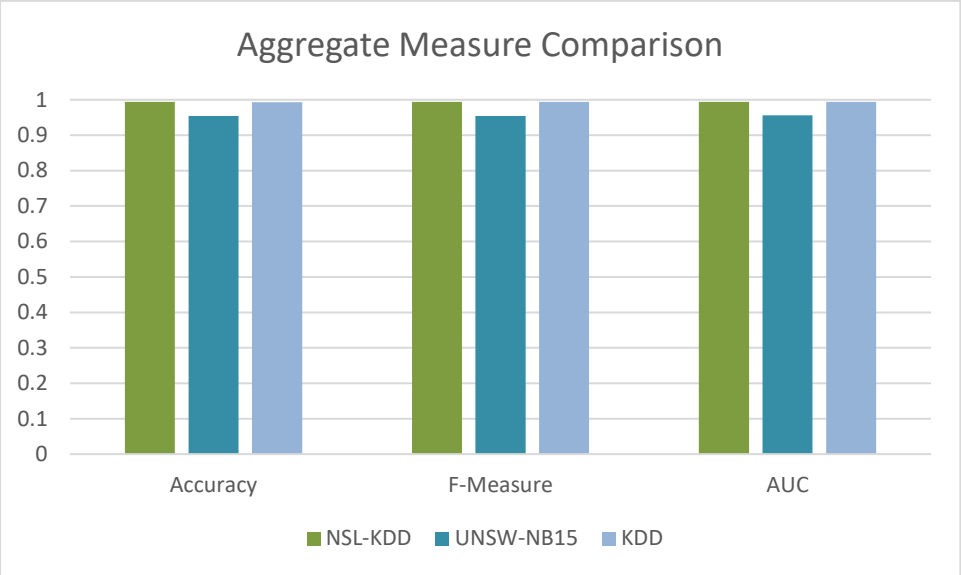

Figure 2: Aggregate Measures of SFS

Table 1 provides a tabulated summary of the performance indicators. The low levels of the fault prediction metrics contrast with the high levels of the positive metrics, demonstrating the strong performance capabilities of the SFS model.

Table 1: Performance Measures of SFS

Technique	NSL-KDD	UNSW-NB15	KDD
FPR	0.0004	0.0815	0.0008
TPR	0.9884	0.9929	0.9884
Recall	0.9884	0.9929	0.9884
Precision	0.9996	0.9185	0.9992
TNR	0.9996	0.9185	0.9992
FNR	0.0116	0.0071	0.0116
Accuracy	0.9936	0.9542	0.9935
F-Measure	0.9940	0.9542	0.9938
AUC	0.9940	0.9557	0.9938

5. Comparative Analysis

The SFS model has been compared to the SAVAER-DNN [25] model. Figures 3 and 4 demonstrate an analysis based on the ROC plot of the NSL-KDD and UNSW-NB15 datasets. The ROC plot of NSL KDD data reveals that the SFS model has the highest true positive and lowest false positive levels. An optimal classifier model has high true positive levels and low positive levels. When comparing the SFS model to the SAVAER-DNN model, the SFS model has a higher true positive rate, nearly one, while the SAVAER-DNN model has slightly lower true positive levels. However, when it comes to false positive rates, SFS has nearly no false positives, whereas SAVAER-DNN has a considerably greater number, indicating that the model generates more false alarms.

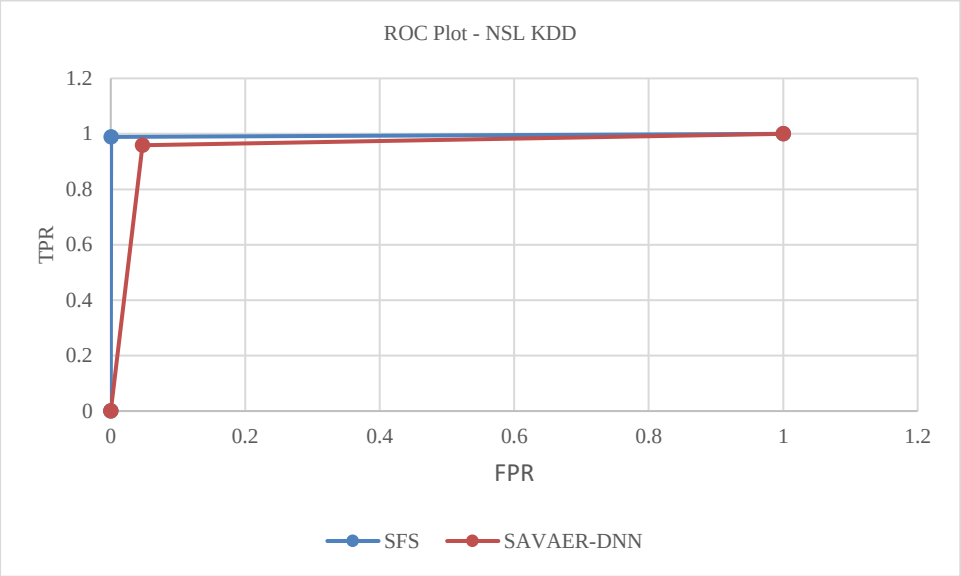

Figure 3: ROC Comparison of SFS on NSL-KDD

Figure 4 displays the receiver operating characteristic (ROC) plot for the UNSW-NB15 dataset. There was an observed false positive rate for both models. When comparing both models' true positive rates, the SFS model outperforms the SAVAER-DNN model.

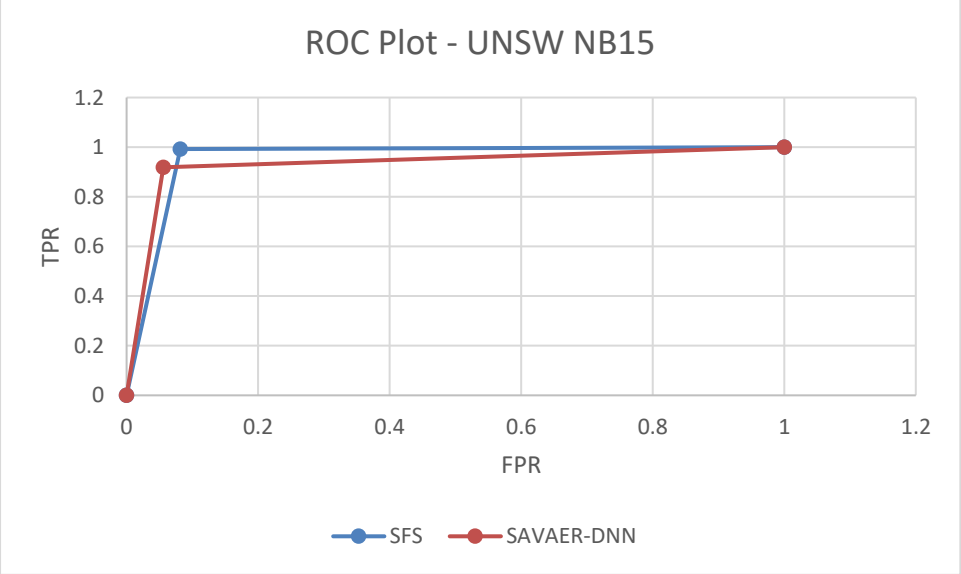

Figure 4: ROC Comparison of SFS on UNSW-NB15

Tables 2 and 3 provide a tabulated breakdown of the results. The most promising forecasts are italicised. On the UNSW-NB15 dataset, the SFS model shows improved prediction performance across the board with the exception of FPR levels. It has been shown that the FPR has decreased by 3%, the TPR has increased by 8%, the accuracy has increased by 2%, and the F-Measure has increased by 2%.

Table 2: Performance Comparison of SFS on UNSW-NB15 Data

	SAVAER-DNN	SFS
FPR	**0.056**	0.08
TPR	0.919	**0.99**
Accuracy	0.930	**0.95**
F-Measure	0.935	**0.95**

The SFS model shows superior prediction performance across all criteria, as shown in Table 3, which is based on an examination of the NSL-KDD dataset. an increase in accuracy by 10% and a decrease in F-measure by 9%; a reduction in FPR by 4%; a rise in TPR by 3%; an F-measure improvement of 9% means the SFS model's forecast is very accurate

Table 3: Performance Comparison of SFS on NSL-KDD Data

	SAVAER-DNN	SFS
FPR	0.047	**0.00042**
TPR	0.959	**0.98838**
Accuracy	0.89	**0.99365**
F-Measure	0.9	**0.99397**

Conclusion

As the number of technologies dependent on networking grows, so does the need for methods that may ensure the security of over-the-peace transfers, especially for sensitive data. Intrusion detection is complicated by data imbalance, which is an inevitable part of any network. In this paper, we describe an intrusion detection architecture that addresses the shortcomings of existing methods by combining a data-balancing module with a prediction module equipped to deal with imbalanced data. Oversampling is employed to maintain data balance in the proposed Stacking and Feature engineering based Semi supervised (SFS) model, which utilises a stacking architecture to combine supervised and semi supervised modelling strategies for prediction. Stacking design addresses the problem of overtraining caused by oversampling. The experimental findings show high performance, with an accuracy of 90% or more across multiple datasets with varying degrees of imbalance. However, on the UNSW-NB15 dataset, the SFS model shows significantly elevated false alarm rates. Some believe this is because of how skewed the data is. Improvements in the future will centre on putting forth a framework for determining imbalance levels and then selecting models accordingly.

References

[1] "Exploiting a crisis: How cybercriminals behaved during the outbreak", https://www.microsoft.com/, 2022. [Online]. Available: https://www.microsoft.com/security/blog/2020/06/16/exploiting-a-crisis-how-cybercriminals-behaved-during-the-outbreak/.

[2] "INTERPOL report shows alarming rate of cyberattacks during COVID-19", Interpol.int, 2022. [Online]. Available: https://www.interpol.int/en/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19.

[3] S. Daneshgadeh Çakmakçı, T. Kemmerich, T. Ahmed, N. Baykal, "Online DDoS attack detection using Mahalanobis distance and Kernel-based learning algorithm", Journal of Network and Computer Applications, vol. 168, p. 102756, Oct.2020..

[4] H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, K.-Y. Tung, "Intrusion detection system: A comprehensive review", Journal of Network and Computer Applications, Vol. 36 (1), pp. 16–24, Jan .2013.

[5] T.F. Lunt, "A survey of intrusion detection techniques", Computers and Security, Vol. 12, Issue 4, pp. 405–418, 1993.

[6] S. Brown, J. Gommers, O. Serrano, "From cyber security information sharing to threat management", in: Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security, Vol.1 pp. 43–49, 12-16 Oct.2015.

[7] L. Dandurand, O.S. Serrano, "Towards improved cyber security information sharing", in: 2013 5th International Conference on Cyber Conflict (CYCON 2013), Vol.1, pp. 1–16, 04-07 Jun.2013.

[8] Levi Victor, Williamson Gillian, King James, "Development of GB distribution networks with low carbon technologies and smart solutions: Scenarios and results", International Journal of Electrical Power Energy Systems. Vol.119, P.105832, July.2020.

[9] Javaid N, Jan N, Javed MU, "An adaptive synthesis to handle imbalanced big data with deep siamese network for electricity theft detection in smart grids", Journal of Parallel and Distributed Computing Vol.153, pp.44–52, Jul.2021.

[10] A. Kumar, K. Abhishek, M. Ghalib, A. Shankar, X. Cheng, "Intrusion detection and prevention system for an IoT environment", Digital Communications and Networks, vol. 8, no. 4, pp. 540-551, Aug.2022.

[11] L. Gui, W. Yuan, F. Xiao, "CSI-based passive intrusion detection bound estimation in indoor NLoS scenario", Fundamental Research, Vol.2,Issue 4, pp.56-65, May.2022

[12] Y. T, G. Murtugudde, "An efficient algorithm for anomaly intrusion detection in a network", Global Transitions Proceedings, vol. 2, no. 2, pp. 255-260, Nov.2021

[13] G. Baldini, I. Amerini, "Online Distributed Denial of Service (DDoS) intrusion detection based on adaptive sliding window and morphological fractal dimension", Computer Networks, vol. 210, p. 108923, Jun.2022.

[14] S. Behal, K. Kumar, M. Sachdeva, “D-FACE: An anomaly based distributed approach for early detection of ddos attacks and flash events”, Journal of Network and Computer Applications, Vol.111, pp. 49–63, Jun. 2018.

[15] J.-H. Jun, C.-W. Ahn, S.-H. Kim, “DDoS attack detection by using packet sampling and flow features”, in: Proceedings of the 29th Annual ACM Symposium on Applied Computing, pp. 711–712, 24-28 Mar.2014.

[16] J. David, C. Thomas, “DDoS attack detection using fast entropy approach on flow-based network traffic”, Procedia Computer Science, Vol. 50 (4), pp.30–36, Jun.2015.

[17] M. Guarascio, N. Cassavia, F. Pisani and G. Manco, "Boosting Cyber-Threat Intelligence via Collaborative Intrusion Detection", Future Generation Computer Systems, vol. 135, pp. 30-43, Oct.2022.

[18] B. Jordan, R. Piazza, T. Darley, “Structured Threat Information Expression™ version 2.1 committee specification 01”, pp.1-313,June.2020.

[19] T. Darley, I. Kirillov, R. Piazza, D. Beck, “Cyber Observable Expression Cybox™version 2.1.1. part 01:Overview - committee specification draft 01”,Vol. 01,pp.1-300,Jun. 2016.

[20] R. Prasad and S. shankar, "secure intrusion detection system routing protocol for mobile ad-hoc network", Global Transitions Proceedings, Vol.4 ,pp.1-11, Oct.2022.

[21] F. Siddiqui, J. Beley, S. Zeadally, G Braught, “Secure and lightweight communication in heterogeneous IoT environments”, Internet Things (2019), Vol.14, P. 100093,Jun. 2021 .

[22] A. Mehmood, A. Khanan, M. M. Umar, S. Abdullah, K. A. Z. Ariffin, H. Song, "Secure Knowledge and Cluster-Based Intrusion Detection Mechanism for Smart Wireless Sensor Networks," in IEEE Access, vol. 6, pp. 5688-5694, Nov.2018

[23] Y. Yang, K. Zheng, B. Wu, Y. Yang, X. Wang, "Network Intrusion Detection Based on Supervised Adversarial Variational Auto-Encoder With Regularization," in IEEE Access, vol. 8, pp. 42169-42184, Feb.2020

[24] M. Asif, S. Abbas, M. Khan, A. Fatima, M. Khan and S. Lee, "Map Reduce based intelligent model for intrusion detection using machine learning technique", Journal of King Saud University - Computer and Information Sciences, vol. 2, pp. 1-11, Dec.2021.

[25] A.Sagaya Priya, S.Britto Ramesh Kumar, “Intrusion Detection using Attribute Subset Selector Bagging (ASUB) to Handle Imbalance and Noise”, International Journal of Computer Science and Network Security, Vol.22, pp.97-102, May.2022.

[26] M. Asif, S. Abbas, M. Khan, A. Fatima, M. Khan and S. Lee, "MapReduce based intelligent model for intrusion detection using machine learning technique", Journal of King Saud University - Computer and Information Sciences, vol. 2, pp. 1-11, Dec.2021.

[27] J.Vimalrosy, S.Britto Ramesh Kumar, "OSS-RF:Intrusion Detection Using Optimized Swarm Based Random Forest Classifier on UNSW-NB15 DATASET", International Journal on “Technical and Physical Problems of Engineering”(IJTPE), Vol.14,Issue 51, pp.275-283, Jun.2022

[28] A. Krari, A. Hajami, E. Jarmouni, "Study and Analysis of RPL Performance Routing Protocol Under Various Attacks" International Journal on “Technical and Physical Problems of Engineering”(IJTPE).Vol.13, Issue 49, pp.152-161, Dec.2021